



# Eine Integrationsmethode zur Umstellung von geodätischen Fortran-Programmen auf Fensterumgebungen

Jürgen Friedrich

Viele Computer-Programme mit geodätischer Zielsetzung für z.B. Ausgleichsrechnungen, Erdzeitenberechnungen, GPS oder VLBI, sind und werden in Fortran geschrieben, doch aufgrund dessen bleibt das Potential von Fensterumgebungen (z.B. Linux oder MS Windows) weitestgehend ungenutzt. Dieses Potential beinhaltet vor allem Möglichkeiten zur dynamischen Speicherplatzverwaltung von Matrizen und Vektoren, flexible Datenein- und -ausgabe sowie die direkte graphische Darstellung von Ergebnissen. Für diese Problematik gibt es zwei mögliche Lösungen: Übertragen oder Integrieren. Weil die Übertragung von Fortran-Programmen in eine andere Sprache sehr arbeits- und zeitaufwendig ist, kann eine Integration in die gewünschte Fensterumgebung vorteilhafter sein. Eine solche Umstellungsmethode wird in diesem Beitrag vorgestellt und mit Beispielen aus der Deformationsanalyse von geodätischen Punktnetzen, Satellitenbahnen- und Erdzeitenberechnungen veranschaulicht.

## 1 Einführung

Diese Arbeit beschreibt eine Methode zur Integration von geodätischen Fortran-Programmen in eine gewünschte Fensterumgebung mittels eines interaktiven und visuellen Shell-Programmes. Solch ein Programm besitzt den Vorteil, dass es die Übertragung von Fortran-Programmen in eine andere Sprache überflüssig macht und die Investitionen in diese Programme erhalten bleiben (Coyle 2000). Es ist eine hybride Methode, die visuelle Interaktivität von Fensterumgebungen und die Qualität von Fortran-Programmen miteinander verbindet. Doch auch Fortran-Programme haben viele Einschränkungen. Z.B. besitzen sie:

- 1) keine dynamische Speicherplatzverwaltung,
- 2) keine Flexibilität bei der Datenein- und -ausgabe, oder
- 3) keine eingebauten Zeichenelemente zur graphischen Darstellung von Ergebnissen.

Falls sich an Punkt 1) nichts ändern lässt, können doch wenigstens die beiden anderen Punkte mit der hier vorgestellten Methode angegangen werden. Darüber hinaus besitzt ein solcher Ansatz noch andere Vorteile. Z.B. lassen sich damit verschiedene Auswertemodelle und Datensätze wie folgt miteinander vergleichen:

- 1) eine Analyse verschiedener Datensätze mit dem gleichen Auswertemodell,
- 2) eine Analyse verschiedener Auswertemodelle mit dem gleichen Datensatz, und

3) eine Analyse verschiedener Auswertemodelle und Datensätze.

Solche Vergleiche sind zur Kontrolle und Bewertung von Datenanalysen und neuen Auswertemodellen unerlässlich. Sie lassen sich ohne große Schwierigkeiten in die hier beschriebene Umstellungsmethode einbauen, und zwar mit einem Multi-File System, das erlaubt, verschiedene Ein- und Ausgabedateien zu öffnen, zu verwalten und zu vergleichen. Seine wesentlichen Bestandteile und ihre Beschreibung befindet sich in Tabelle 1.

Dieses Multi-File System lässt sich einfach mittels eines Multiple Document Interfaces (MDI) in Kombination mit einem Multi-File Management System realisieren, wobei ein Exe-file einen Hauptbaustein darstellt, und *MyFile* ist die Basisklasse in dem benutzten UML Modell. Das entwickelte Shell-Programm läuft unter dem MS Windows 9 x/NT/2k Betriebssystem, und seine Programmierung wurde in Visual Basic (VB) durchgeführt. Alle Dateien inkl. der Beispiele stehen zur Verfügung unter <http://www.simtel.net/pub/dl/58693.html> (3.2 MB). Beispiele mit beigefügten Quellcodes befassen sich mit der

- Deformationsanalyse von geodätischen Punktnetzen,
- Satellitenbahnenberechnung, und der
- Erdzeitenberechnung.

Für Visual Basic gibt es sehr viele Quellen, z.B. Bücher von DEITEL et al. (1999), SCHNEIDER (1998) oder STEPHENS (2000), um nur eine kleine Auswahl zu nennen.



**Tab. 1: Multi-File System für Ein- und Ausgabedateien**

Bestandteil	Beschreibung
Dateinamen (programmiert in „MyName.EXE“)	„MyName.EXE“ liebt „MyName.TXT“, die Ein- und Ausgabedateinamen enthält, die „MyName.INP“ und „MyName.-OUT“ benannt sind.
Mehrfaches Ausführen von „File/Open“ zum Öffnen der gleichen Datei „MyName.EXE“	1 <sup>st</sup> Datei geöffnet heißt „1MyName.TXT“, 2 <sup>nd</sup> Datei geöffnet heißt „2MyName.TXT“, 3 <sup>rd</sup> Datei geöffnet heißt „3MyName.TXT“, ... n <sup>th</sup> Datei geöffnet heißt „nMyName.TXT“, die beinhalten „nMyName.INP“ und „nMyName.OUT“, die auch editiert werden können während des Programmlaufes.
Ausführen von „File/Run“	1. Kopiert „MyName.TXT“ nach „OldMyName.TXT“, 2. Kopiert „nMyName.TXT“ nach „MyName.TXT“, 3. Führt „MyName.EXE“ aus, die „nMyName.INP“ liebt und „nMyName.OUT“ erzeugt, 4. Kopiert „OldMyName.TXT“ nach „MyName.TXT“, 5. Zeigt „nMyName.OUT“ als Text und Plot, 6. Falls $n > 2$ , dann ermögliche die Differenz „n1Name.OUT“ – „n2MyName.OUT“ als Text und Plot.

## 2 Objektorientierte Analyse und Design

Das interaktive Shell-Programm für Exe-Dateien soll den folgenden Anforderungen genügen. Es soll in der Lage sein:

- 1) verschiedene Exe-Dateien gleichzeitig zu verwalten und auszuführen,

- 2) Ein- und Ausgabedateien visuell und interaktiv während des Programmlaufes zu öffnen und zu verändern, und
- 3) Ergebnisse und deren Differenzen graphisch darzustellen.

Das Flussdiagramm für ein Fortran-Programm wird in Abb. 1 gezeigt mit einem Exe-file sowie Ein- und Ausgabedateien genannt Input und Output (I/O) Files.

Ein Exe-file liebt einen Input File, berechnet das Ergebnis und schreibt

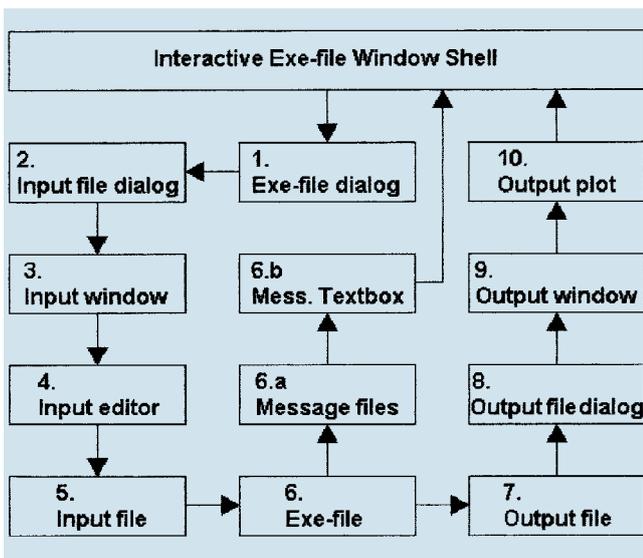
es in einen Output File. Darüber wird das Shell-Programm gelegt, wobei Shell, Exe-file und I/O Files gemäß Abb. 2 miteinander verknüpft sind. Dieses Flussdiagramm kann sofort in das folgende Use-case Diagram (Abb. 3) der Unified Modeling Language (UML) übertragen werden, dem internationalen Standard für objektorientierte Modellbildung (Booch et al. 1998). Die neueste Version und ihre Spezifikationen sind unter <http://www.omg.org/uml/> einzusehen.

Die Grundlage des Shell-Programm Designs bildet eine 3-Tier Architektur mit den folgenden Schichten: 1) Oberste Schicht ein Graphisches User Interface (GUI), 2) Mittlere Schicht ein Prozessor, und 3) Untere Schicht ein Datenmanager. Eine mögliche Realisierung dieser Architektur wird in dem Klassendiagramm in Abb. 4 gezeigt, wo die drei Schichten in die obersten Klassen abgebildet wurden.

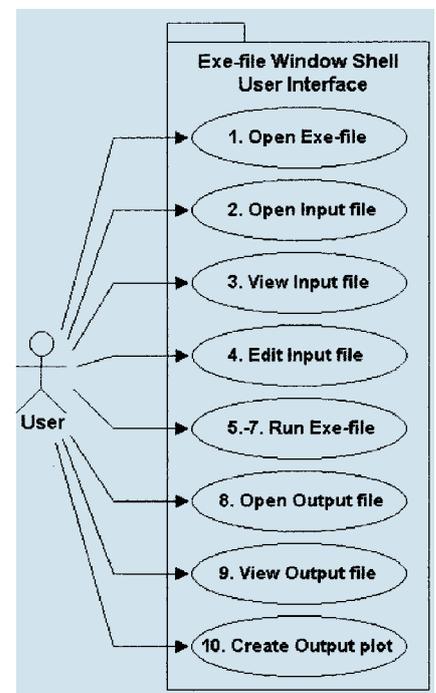
Die Klasse *UserInterface* ist von Klasse *MDIForm* durch eine GenSpec Association abgeleitet, wobei die Klassen *Processor* und *DataManager* eine Whole-to-part Association mit *UserInterface* besitzen. Beide Klassen *Processor* and *DataManager* sind von der Klasse *Module* ab-



**Abb. 1: Flussdiagramm für ein Fortran-Programm mit I/O Files**



**Abb. 2: Flussdiagramm für ein Shell-Programm**



**Abb. 3: Use-case Diagram für ein Shell-Programm mit 1–10 Schritten**

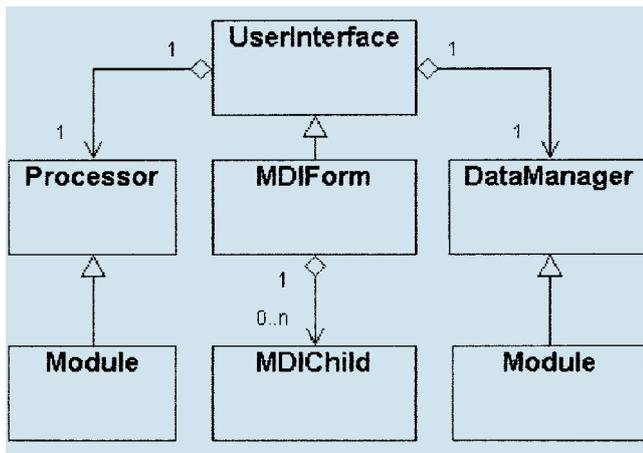


Abb. 4: Klassendiagramm für eine 3-Tier Architektur

geleitet. Die Klasse *MDIForm* (Multiple Document Interface Form) ist der Erzeuger eines Hauptfensters und vieler *MDIChild* Fenster, die u.a. den Inhalt von Dateien darstellen können. Die Klasse *MDIChild* hat eine Gen-Spec Association mit der Klasse *Form*, die zu der Microsoft Foundation Class (MFC) Library gehört, die mit Visual Basic mitinstalliert wird. Ein *MDIForm* Objekt beinhaltet i.d.R. ein Menu, Tool Bar und Status Bar sowie einen Child Window Bereich.

Das Klassendiagramm des Shell-Programms baut auf Abb. 4 mit anderen zusätzlichen Klassen auf:

- Klasse *CommonDialog* für die Auswahl von Exe-file und I/O Files,
- Klasse *Collection* für die Verwaltung von Child Fenster, und
- Klasse *MyFile* für Dateiattribute und -operationen.

Alle diese drei Klassen haben eine Whole-to-part Association mit Klasse *UserInterface* (*MyFile* durch *MDIForm*). Klassen *CommonDialog* und *Collection* stammen auch aus der MFC Library, wogegen Klasse *MyFile* selbst geschrieben ist, deshalb die Vorsilbe „My“.

Andere zusätzliche Klassen sind:

- Klasse *MyPoint* zur Erzeugung von Datenpunkten inklusive ihrer Koordinaten,
- Klasse *PointList* zur Verwaltung von *MyPoint* Objekten, und
- Klasse *CheckBox* zur Auswahl und Kontrolle von Plotparametern.

*MyPoint* ist eine selbstgeschriebene Klasse mit ID und Xi,Yi Koordinaten als Attribute. *PointList* and

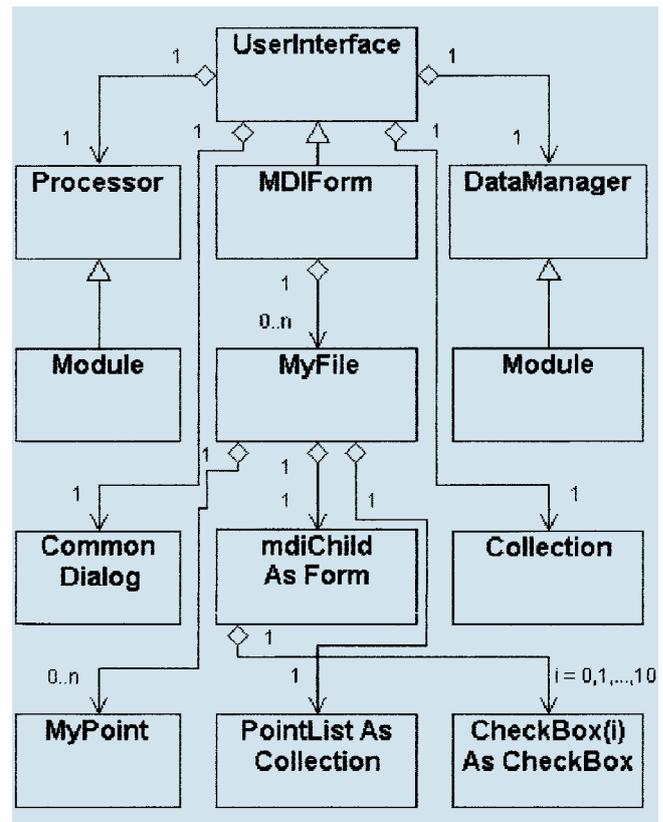


Abb. 5: Klassendiagramm für das Shell-Programm

*CheckBox(i)*,  $i = 0,1, \dots, 10$ , besitzen eine Gen-Spec Association mit Klassen *Collection* and *CheckBox*, die ebenfalls der MFC Library entnommen sind. Mit all diesen Klassen ergibt sich daraus das Klassendiagramm in Abb. 5.

Das entsprechende UML Sequence Diagram für den Use-case in Abb. 3 sieht wie folgt aus (Abb. 6). Das Design eines *mdiChild* Fensters beruht auf einer Tab Klasse *SSTab*, die mit *mdiChild* über eine Whole-to-part Association verknüpft ist.

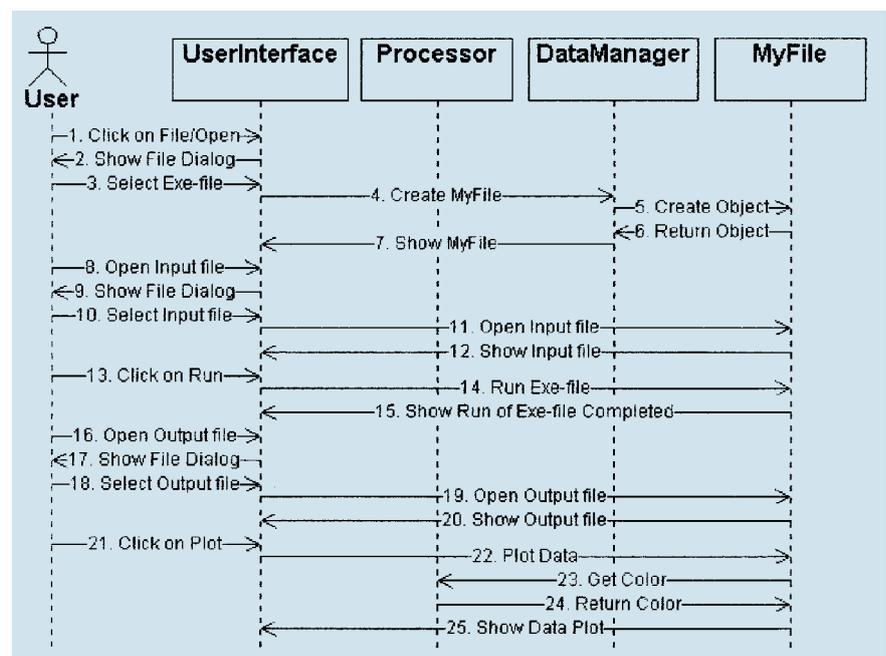


Abb. 6: Sequence Diagram für das Shell-Programm

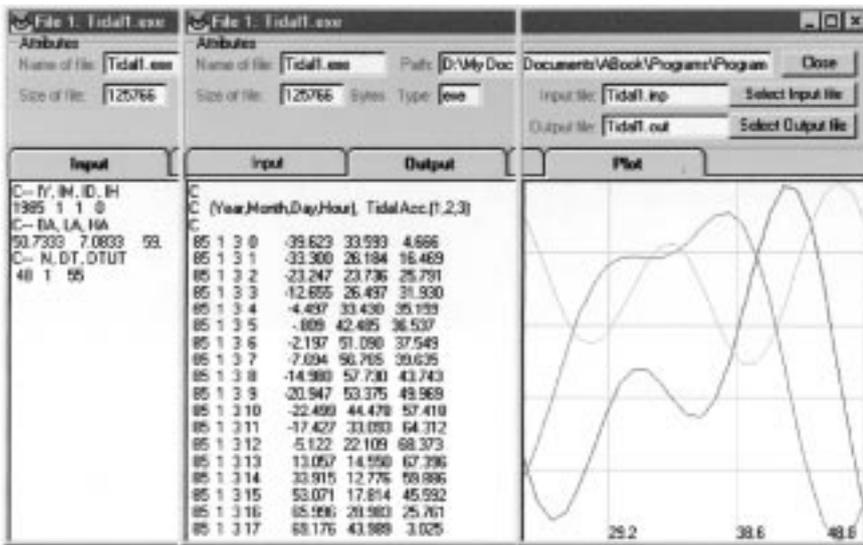


Abb. 7: Ein mdiChild Fenster mit einem SStab Objekt als Teil des Shell-Programmes

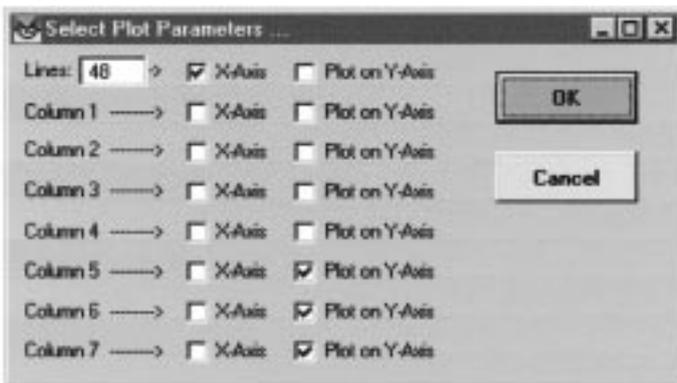


Abb. 8: CheckBox(i), i = 0, 1, ..., 10 Objekte als Teil eines frmSelect Fensters

SStab ist auch der MFC Library entnommen und wurde nicht in das Klassendiagramm (Abb. 5) aufgenommen, um es nicht unnötig mit Informationen zu überladen. Ein SStab Objekt innerhalb eines mdiChild Fensters besitzt drei Tabs genannt „Input“, „Output“ and „Plot“ (Abb. 7). Jedes Tab ist mit folgenden Funktionen verbunden. Ein Mausklick auf:

- Tab „Input“ zeigt den Inhalt der Eingabedatei in einem Textfenster an, der hier editiert werden kann, bevor er gespeichert und vom Exe-file gelesen wird.
- Tab „Output“ zeigt den Inhalt der Ausgabedatei in einem Textfenster an, so wie er vom Exe-file erzeugt wurde.
- Tab „Plot“ zeigt den Inhalt der Ausgabedatei als 2D Plot an, wobei die Ausgabedaten in einem Tabellenformat vorliegen sollten, so wie im Output Fenster von Abb. 7.

In Abb. 8 beinhaltet die Ausgabedatei sieben Datenspalten, doch nur die letzten drei Spalten sind als Graphik in Abb. 7 gezeichnet worden. Die Auswahl der Spalten für das Plotfenster geschieht über die CheckBox(i) Objekte (Abb. 5) als Teil eines frmSelect Fensters, so wie es in Abb. 8 gezeigt wird.

### 3 Implementierung des Designs

Die Implementierung und Programmierung des Shell-Programmes kann sehr übersichtlich mit Hilfe des VB Projekt Explorers (McMonnies 2001) dargestellt werden. Abb. 9 wird sofort sichtbar, nachdem die Projektdatei „MyShell.vbp“ im Ordner „MyShell“ mit Visual Basic Version 5 oder 6 geöffnet worden ist. Der ganze Quellcode mit allen

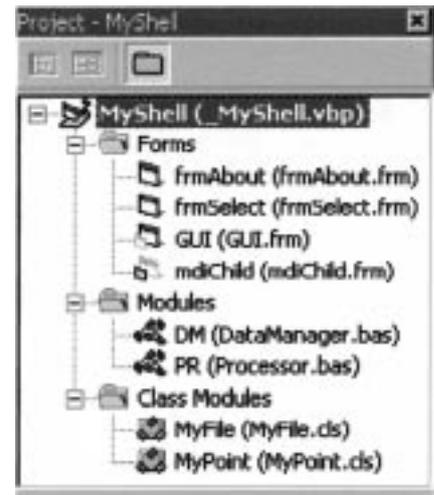


Abb. 9: VB Project Explorer Fenster für das Shell-Programm

Beispielen ist verfügbar bei <http://de.geocities.com/bsttc3/dload/MyShell.zip> (läuft mit VB Version 5 oder 6).

Wie das ganze Projekt verwirklicht wurde und wie das Shell-Programm genau funktioniert, lässt sich am besten mit dem Sequence Diagramm (Abb. 6) erläutern, wenn man jeden Schritt und den dazugehörigen Code einzeln erklärt. Weil das aber den Rahmen dieses Beitrages sprengen würde, soll an dieser Stelle nur ein kleines Beispiel beschrieben werden. Wenn der Exe-file durch einen Klick auf „File/Run“ in der Menu Bar aufgerufen wird, wird der folgende Programmcode ausgeführt, wie er in Tab. 2 aufgelistet ist. Nachdem das aktive mdiChild Fenster mit der Tatsache gefunden wurde, dass das Attribut „FormID“ eines mdiChild Fensters identisch mit dem Attribut „FileID“ eines MyFile Objektes ist, wird die Operation „RunExeFile“ eines MyFile Objektes aufgerufen, was dann den Input File liest, das Ergebnis berechnet und in den Output File schreibt.

### 4 Tests und Beispiele

Das Shell-Programm wurde schließlich in verschiedenster Weise getestet, so dass alle Anforderungen gemäß Abschnitt 2 erfüllt werden. Dabei wurde die meiste Zeit für die Fehlerbeseitigung der „RunExeFile“ Operation eines MyFile Objektes verwendet.

Tab. 2: Kode für „File/Run“ in Datei „GUI.frm“ in Ordner „MyShell“

	...
01	Private Sub mnuRun_Click()
02	Dim obj
03	For Each obj In ChildList ' Find child in list
04	If val(Screen.ActiveForm.FormID) = val(obj.FileID) Then
05	Screen.MousePointer = vbHourglass
06	Call obj.RunExeFile
07	Screen.MousePointer = vbDefault
08	End If
09	Next obj
10	End Sub
	...

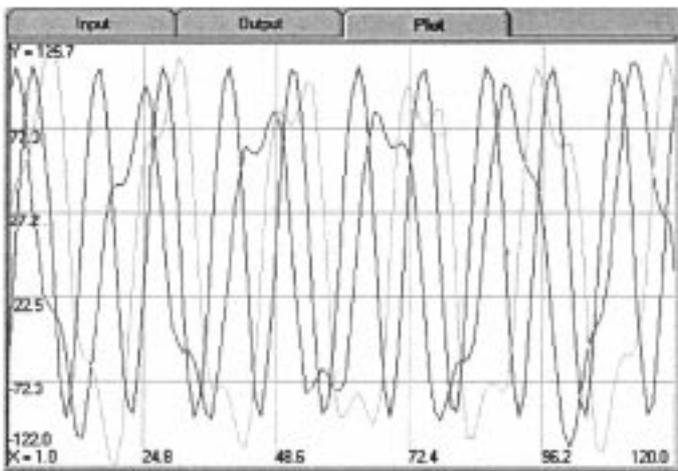


Abb. 10: Kartesische Bahndifferenzen in Metern für zwei GPS-Satelliten über 5 Tage

Wie bereits erwähnt wurde, sind folgende geodätische Anwendungen zum Testen benutzt worden:

- Deformationsanalyse von geodätischen Punktnetzen,
- Satellitenbahnenberechnungen, und
- Erdzeitenberechnungen.

Alle Dateien inkl. der Quellcodes stehen zur Verfügung unter der folgenden Adresse: <http://www.simtel.net/pub/dl/58693.html> (3.2 MB). Als Beispiel wurde Abb. 10 beigelegt, die kartesische Bahndifferenzen in Metern für zwei GPS-Satelliten mit einer Perigäumdifferenz von einer Bogensekunde über eine Periode von 5 Tagen zeigt.

## 5 Schlussfolgerungen

Das Ziel dieses Beitrages war die Beschreibung einer Methode zur In-

tegration von geodätischen Fortran-Programmen in eine gewünschte Fensterumgebung mittels eines interaktiven und visuellen Shell-Programmes. Zahlreiche Tests und verschiedene Anwendungen am Institut des Verfassers in den Bereichen Deformationsanalyse von geodätischen Punktnetzen, Satellitenbahnen- und Erdzeitenberechnungen haben die Tauglichkeit der Methode untermauert. Praktische Erfahrungen haben gezeigt, dass diese Methode zwei Hauptbeschränkungen besitzt. Erstens bedarf es Exe-Dateien, die alleine im stand-alone Modus ausgeführt werden können, ohne den zur Verfügung stehenden Speicher zu überschreiten. Zweitens stoßen die graphischen Darstellungsmöglichkeiten der Ergebnisse bald an ihre Grenzen. Zwar können ohne viel Aufwand z.B. Verschiebungsvektoren und Konfidenzellipsen in einen

Punktplan mitgezeichnet werden, aber alles, was darüber hinausgeht, lohnt vermutlich nicht den Aufwand und kann wahrscheinlich mit anderen Graphikprogrammen einfacher dargestellt werden.

## Literatur

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I.: (1998) Unified Modeling Language User Guide, Addison-Wesley Massachusetts

COYLE, F. P.: (2000) Legacy Integration – Changing Perspectives, IEEE Software March/April pp. 37–41

DEITEL, H. M.; DEITEL, P. J.; NIOTO, T.: (1999) Visual Basic 6 How to program, Prentice-Hall New Jersey

MCMONNIES, A.: (2001) Visual Basic – an object oriented approach, Addison-Wesley Massachusetts

SCHNEIDER, D. I.: (1998) An Introduction to Programming Using Visual Basic 5, 3rd ed., Prentice-Hall New Jersey

STEPHENS, R.: (2000) Visual Basic Graphics Programming – Hands-On Applications and Advanced Color Development, John Wiley & Sons Chichester UK

Adresse:

Asst. Prof. Dr. JÜRGEN FRIEDRICH, Dept. of Geodesy & Photogrammetry, Karadeniz Technical University, KTU Lojman 35/7, 61080 Trabzon, Turkey Tel: + 90-(0) 4 62-3 77 27 62, Fax: + 90-(0)4 62-3 25 79 77 or 3 28 09 18, E-mail: jafriedrich@yahoo.de

## Zusammenfassung

Viele Computer-Programme mit geodätischer Zielsetzung für z.B. Ausgleichsrechnungen, Erdzeitenberechnungen, GPS oder VLBI, sind und werden in Fortran geschrieben, doch aufgrund dessen bleibt das Potential von Fensterumgebungen (z.B. Linux oder MS Windows) weitestgehend ungenutzt. Dieses Potential beinhaltet vor allem Möglichkeiten zur dynamischen Speicherplatzverwaltung von Matrizen und

**Vektoren, flexible Datenein- und -ausgabe sowie die direkte graphische Darstellung von Ergebnissen. Für diese Problematik gibt es zwei mögliche Lösungen: Übertragen oder Integrieren. Weil die Übertragung von Fortran-Programmen in eine andere Sprache sehr arbeits- und zeitaufwendig ist, kann eine Integration in die gewünschte Fensterumgebung vorteilhafter sein. Eine solche Umstellungsmethode wird in diesem Beitrag vorgestellt und mit Beispielen aus der Deformationsanalyse von geodätischen Punktnetzen, Satellitenbah-**

**nen- und Erdzeitenberechnungen veranschaulicht.**

#### **Abstract**

**Many computer programs for geodetic purposes like those for adjustments, earth tide computation, GPS or VLBI were and will be written in Fortran, though the potential of window platforms (e.g. Linux or MS Windows) remains mostly unused. This potential contains possibilities for dynamic memory management of matrices and vectors, flexible data in- and output as well as a direct graphical repre-**

**sentation of results. For this problem two possible solutions are available: migration or integration. Because migration of Fortran programs to another language is very work-intensive and time-consuming, the integration into the wanted window platform can be more advantageous. Such an integration method is introduced in this paper and illustrated with examples taken from the deformation analysis of geodetic networks, satellite orbit determination and earth tide computation.**

## **Geodäsie in Kaliningrad**

Am 5. Dezember 2001 wurde die neue Produktionsbasis des Aero-geodätischen Betriebes in Kaliningrad (dem früheren Königsberg) in Anwesenheit bedeutender Persönlichkeiten – darunter der Fliegerkosmonaut der UdSSR und zweifache Held der Sowjetunion, Rektor der Moskauer Universität für Geodäsie, Luftbildaufnahme und Kartografie (Migaik) Prof. Dr. techn. Savinyj, und der Vorsitzende des Rates der Betriebsdirektoren von Roskartografija und Direktor des Kartgeozentrums Berk – feierlich eröffnet.

In seiner Rede erinnert der stellvertretende Leiter des Föderalen Dienstes für Geodäsie und Kartografie Russlands Prusakov an die Schwierigkeiten, als 1977 die ersten Geodäten und Kartografen mit dem Aufbau der Organisation in Kaliningrad begannen, in dieser entfernten russischen Enklave, getrennt vom Hauptgebiet des Landes, jetzt umgeben von einigen unabhängigen Staaten. Inzwischen weist diese Einrichtung mit 2500 m<sup>2</sup> eine moderne Ausstattung mit einer unikalen Herstellungslinie für digitale

Karten, mit 49 PC hoher Produktivität und weiteren modernen Produktionsmitteln auf. Damit besteht Gewissheit, dass die Herstellung digitaler Karten 1:10 000 des Kaliningrader Gebiets als Grundlage eines automatisierten Katasters und für Aufgaben der GIS erfolgreich gelöst werden kann. Als erste Region in Russland ist diese so ausgerüstet, dass sie sich mit Deutschland vergleichen kann, wo solche Ausrüstungen für das gesamte Staatsgebiet vorhanden sind. Die Druckerei arbeitet rund um die Uhr und stellt fünf Exemplare Karten her. – Wie im gesamten Gebiet

Russlands wird auch hier die Ausgleichung der trigonometrischen Netze 3. und 4. Ordnung abgeschlossen. In Russland werden bis zum 1. Juli 2002 3800 Kataloge in je 150 Exemplaren vorliegen, die die geodätischen Punkte im Koordinatensystem SK-95 aufweisen. Aus: V dobryj put', Balt AGP. Von Ševnja, M. S. – Geodez. i. Kartogr., Moskva (2002) 2, S. 34–38.

DEUMLICH

